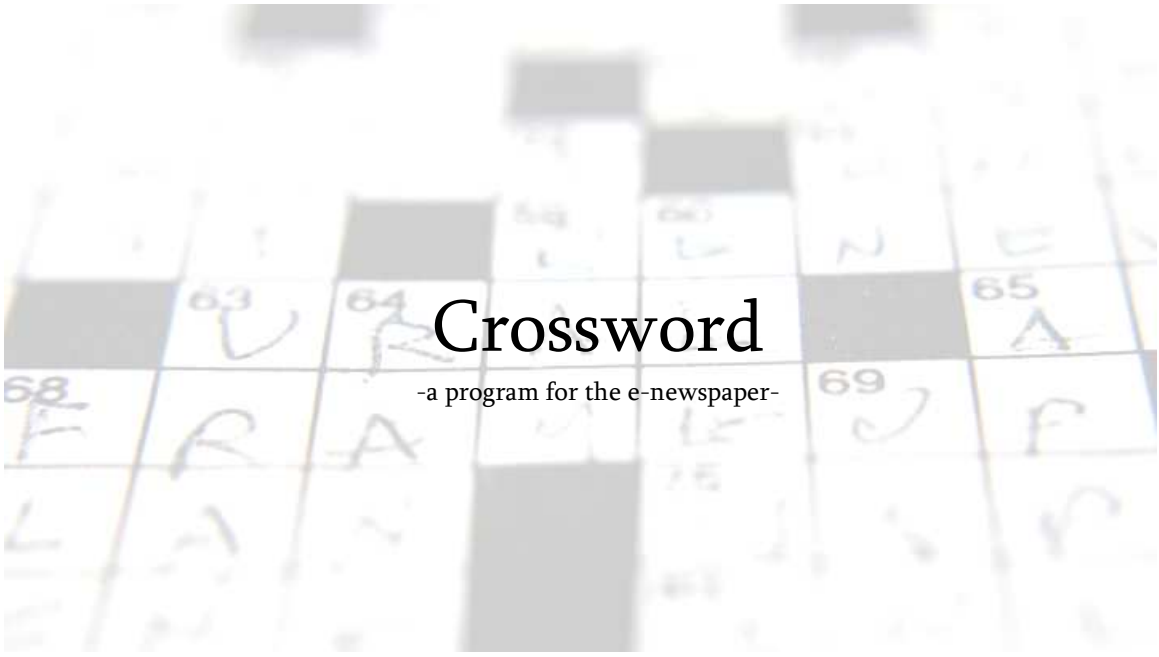


Crossword

-a program for the e-newspaper-

Crossword

-a program for the e-newspaper-

Author:
Giel de Steur

Date:
29 May 2006

Preface

Within an exchange project between Halmstad University and Fontys Hogescholen in Eindhoven (the Netherlands) I was able to do a part of my Dutch study Men & Information technology in Sweden. In this context I designed and built one new service, which was not yet implemented in the e-newspaper (a digital newspaper); a crossword puzzle. This project was one of the two I had to complete during the exchange period of 100 days.

This project was carried out as a part of the DigiNews. DigiNews is a project where Media-IT participates in. I chose to do the project in this context because DigiNews already had produced many results. There was a framework for an e-newspaper and now the extra services were becoming important. The services variated and I specifically chose for the crossword puzzle because that is the service that has a long history with the newspaper and is used by many people. The crossword puzzle was a challenge and it was also possible to complete the project within the time I had.

This report is primarily written for my supervisors. Next to that I think this report can be valuable for other students that want to know more about designing software or that are working with (design of) e-newspaper services. The report itself gives insights in how the process in this project elapsed, next to that it describes the results of this project and comes up with design guidelines.

At last I want to thank the MI-lab for giving me the possibility to take place in this research. I also want to thank the staff members and students that were very kind and always helpful. They made this project a success and gave me many funny moments and a lot of pleasure, next to that they gave me useful (critical) comments and lots of information. Especially I want to thank Jesper Svensson for the great supervising.

Halmstad, Sweden, June 2006

Table of contents

Summary

This project was launched to design and built a service for an e-newspaper. In this project one service was chosen; the crossword puzzle. The central question in this project was:

"How does one design and build a crossword puzzle, that will be shown on an e-reader device and that meets the user-requirements of e-newspaper readers?".

This question implies that the user-requirements for this project were not yet known and had to be find during the project. This was one of the main reasons for chosing the software development method for this project. In this project user centered design was used. User centered design is a software development method that involves the user in almost every stage of the design process. The output should be a product that is tailor made for the user.

To develop a view of who belongs to the group e-newspaper readers and how they use the crossword puzzle different techniques were used. First an analyse was made of the user group. The user group contains a very wide spectrum of people. In this research e-newspaper readers were devided into three categories. Category A were people in the age from 15 to 30. Category B were people in the age from 30 to 65 and the last category, C contained people over 65. After the knowledge of who those people were the crossword puzzle was analyzed. A task flow was made, this gave insight on how a crossword puzzle was filled in. After the task flows, personas were made. The constructing of personas is a technique where the developer writes about imaginary people. This helps the designer to get replace himself in them. Scenarios were also made to imagine how those personas would interact with the crossword puzzle.

After having this theoretic framework the first prototype was developed. This was a low-fidelity prototype. Four sketches were made on paper and different color schemes were added. The user commented the prototypes and the first set of user requirements were found.

The user comments on the low-fidelity prototype were being used to design and built a high-fidelity prototype. This high-fidelity prototype was a fully functional version of the application. The user commented this prototype as well and the second set of user requirements was found.

After all user input the high-fidelity prototype was converted into the final solution. This final solution was based on the new user requirements and the comments as found during the evaluation. The final solution is made with the programming language Java. Java is an outstanding tool for building applications, from large application to small applications, from much interactivity to less interactivity. The application was packed in an installer program to make it easier for the user to install the program.

The final user requirements found during the project were:

UR1. Don't place to many components on the screen;
UR2. What should been seen first should be placed left;
UR3. Use cool colours instead of warm colours;
UR4. Built the application on the users' existing mental model.

UR5. Use well readable lettertypes;
UR6. Things that are crucial for the application should not disappear from the screen and it should not be possible to erase them either.
UR7. State that items in a list are clickable;
UR8. Do not make one button for two functions. Use two different buttons instead.

1. Backgrounds

In this chapter the reader will be provided with a look on the backgrounds belonging to this problem. The backgrounds are especially interesting to get a good and real look on the problem as it will be described later in paragraph 2.1. In this chapter there will be started with a look on the history of the crosswordpuzzle in general (§1.1), after that there will follow a short introduction to what e-paper is (§1.2) and then a profile of the company that is facing the problem (§1.3) will be provided.

1.1 History of the crosswordpuzzle

Nowadays we can find a crosswordpuzzle in almost every newspaper around the world, at least in the paper versions. There is a need to implement crosswordpuzzles also in the e-newspaper. The history of the crossword puzzle is intertwined with the paperversion of the newspaper.

The history of the crosswordpuzzle is heading back to the 19th century where the first crosswords appeared in England (Cambon Jensen, 1997). They were of an elementary kind, apparently derived from the word square, a group of words arranged so the letters read alike vertically and horizontally, and printed in children's puzzle books and various periodicals. However it was not England, but the United States where the crossword puzzle became so populair as it is now (crossword puzzles are said to be the most popular and widespread word game in the world).

The first published crossword puzzle apperead in a Sunday newspaper, the New York World on December 21, 1913. It was created by a journalist anmed Arthur Wynne born in Liverpool. Wynne's puzzle differed from today's crosswords in that it was diamond shaped and contained no internal black squares. During the early 1920's other newspapers picked up the newly discovered pastime and within a decade crossword puzzles were featured in almost all American newspapers. It was in this period crosswords began to assume their familiar form. Ten years after its rebirth in the States it crossed the Atlantic and re-conquered Europe.

The first appearance of a crossword in a British publication was in Pearson's Magazine in February 1922, and the first Times crossword appeared on February 1, 1930. British puzzles quickly developed their own style, being considerably more difficult than the American variety. In particular the cryptic crossword became established and rapidly gained popularity.

Since December 21, 1913 it is common believe that every newspaper around the world has a (crossword) puzzle. For almost a century now crosswords were published in newspapers (Cambon Jensen, 1997), the crossword puzzle became popular due to this media and newspaper producers see the crossword puzzle as necessary for their newspaper.

1.2 E-newspaper

Electronic paper (e-paper) is the common term for a multitude of different technologies that can be used to produce screens with a number of specific characteristics. The e-paper is reflecting, giving the same reader experience as paper (such as high contrast and the possibility to read in sunlight). The paper is thin, flexible and non-sensitive. In addition, it does not require high battery performance (Ihlström, 2005).

E-paper technology is developing fast. More and more companies invest in "plastic electronics" and come up with new prototypes or launch their products on the consumer market. Nowadays for example Sony, Philips, iRex, Fujitsu have products or prototypes for consumer use. The prototypes differ in application. Philips for example developed a rollable display, in that prototype the screen can roll up and has a bending rate of 0,75 cm. Another example is Fujitsu, who has a bendable color display. The e-paper technique is initially meant for

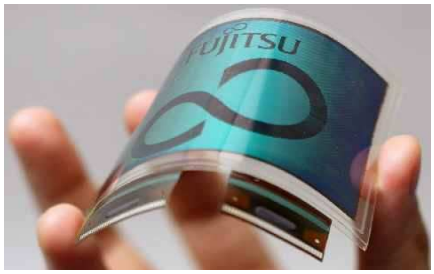


Image 2: bendable color display

devices on which a user can read text or see images.
The Sony Reader, a new device that is now available in the U.S., is able to perform approximately 7500 page turns per battery change. The Sony Reader is not only applicable for eBooks but it allows users to open more file types, such as PDF files, which can be read (Sony, 2006).
iRex Technology BV is a spin-off from Royal Philips Electronics. In April 2006 they made their e-reader device called the "iLiad" available on the Dutch market (iRex, 2006). The iLiad has sufficient memory for 1 month of newspapers, 30 books and many other documents. The user can interact with the device using the stylus, a sort of pen. This device has, upon other ways the possibilities to connect via Wi-Fi (802.11g).

All those devices are fully mobile and consume very less battery power which improves the portability. This mobility is exactly what the consumer wants these days. Our society gets more and more mobile. People travel far more then fifty years ago and there is a need for information anywhere, anytime (Kristoffersen and Ljunberg, 2000).

This need for information is involving the media. Media changes constantly, not always visibly but examples are the coming of the FM radio, television and the printing press. In today's (western) society much information is digital. The media took a small step into this digital world by introducing news on internet. The next step is expected to be news on mobile devices. Negroponte (1995, p. 152) already envisioned this: *"Imagine an electronic newspaper delivered to your home as bits. Assume it is sent to a magical, paper-thin, flexible, waterproof, wireless, lightweight, bright display. The interface solution is likely to call upon mankind's years of experience with headlining and layout, typographic landmarks, images and a host of techniques to assist browsing"*.

If one considers this new development in IT, the change of the media and the fact that people get more and more mobile it is possible to conclude that these three things together pave a road for a development of an e-newspaper. An e-newspaper is a newspaper that is published on a device as mentioned above. It takes the benefits from the newspaper on paper (such as mobility, reading in sunlight, etc.) and the benefits from IT (e.g. interactivity and fast and easy updates).

1.3 Company profile

In this paragraph the reader will be explained for which company this project is. It will give an overview on how the company is organised, what targets the company has, how they finance their activities and it will generally explain an important project.

The research group for which the crossword project has been launched is called Media-IT. It is a young research group; they started about three years ago. Media-IT is a self-employed part of the "Informationsvetenskap, Data- och elektroteknik" department of "Högskolan i Halmstad". They work in projects which are carried out in cooperation with other companies (which can benefit from the research results).
The Media-IT group employes about 6 people (some parttime and some fulltime) for Media-IT. The research group is managed by Carina Ihlström, the program manager. Below that there are Phd. Students, who do research and take part in guiding students from Högskolan i Halmstad. The research group makes much use of students, who help in projects as part of their education.

As the name of the research group already points out the field of work is a combination of Media and IT. The research group is interested in new forms of media (e.g. electronic newspapers) which can be deployed by using Information Technology.

The most important target from the Media-IT research group is to develop new knowledge which is usefull to other companies and the research community. They use new techniques and new devices; they try to develop them according to the target of the projects. The other companies work in close cooperation with Media-IT. This cooperation comes up with new knowledge, the company, Media-IT and the research community benefits from it. This upwards spiral gives a huge push forward to gaining new knowledge.

The Media-IT group is a non-profit organisation, mostly financed by national research boards. Media-IT send in applies for projects and money can be granted by national research boards in Sweden, like ITEA. This way of gaining money also implies that Media-IT is working based on projects. Each project can differ but of course has a relation with Media and IT.

Högskolan i Halmstad provides Media-IT with the necessary offices, office supplies, standard IT equipment, such as computers and students. The extra IT equipment needed is bougth by Media-IT itself. This can ofcourse differ according to the project but examples are tablet PC's, GSM phones and PDA's.

Media-IT has 3 projects at this moment. The first project is relativily small and for the tourism office in Halmstad. The second project is a new project; it is called "Designing Ubiquitous Media Services through Action Research". This project aims at finding solutions for multi-channel publishing of mobile services in a newspaper context, on different devices. The third and last project is called "DigiNews", this project is almost ended now. In two years (2004-2006) Media-IT was aiming at the change from the paper version of the newspaper to a digital newspaper presented on e-paper, combining the best from the printed newspaper with the best of the online newspaper (on mainly the e-reader). The crossword puzzle project fits in the Diginews project, as being a part of the e-newspaper.

2. Problem and target description

In this chapter the problem description (§2.1) and the targets (§2.2) will be presented to you. The problem description and the targets are specified in close consulatition with a staff member within the Media-IT department. The research that is described in the underlaying document is based on this problem description and these targets.

2.1 Problem description

For a project it's important to describe a problem definition. With a problem description all the people involved, and also the reader, will form a picture from what the problem is within the scope of the project.

At this moment the Media-IT department doesn't have enough resources to do a proper investigation on how to optimally design a crossword puzzle for the readers of a (e-) newspaper. It is however remarkable that the crossword puzzle is not yet investigated and build to be shown on e-reader. This particularly considered the interactive potential and the history of the puzzle itself.

This resulted in the following problem description:

"How does one design and build a crossword puzzle, that will be shown on an e-reader device and that meets the user-requirements of e-newspaper readers?".

As you might notice the description is written down like a research question. This makes it easier to specify the goal. In this project the answer to the question has to be found, further on the expectation is that this way of describing is fully functional for the project.

2.2 Targets

The goal for this project is to come to a balanced answer on the question as formulated in paragraph 1.1. Therefore it is needed to answer the following questions:

- Who is part of the group "readers of the e-newspaper"?
- What are the user-requirements of the readers?
- What should be the GUI design based on those user-requirements?
- How is it possible to build that as optimal as possible?

Within the project there will be the design and build of the prototypes and the final version, as being the softwareproducts that contain the crossword puzzle. Of course all of this needs to be well documented.

3. Preparation

In this chapter the preparation phase will be described. The preparation phase is nessecary to plan the project. In paragraph 3.1 the software development method will be described with arguments why the chosen method fits the project. In the last paragraph there will be an overview from the tools that will be needed in order to complete the project (§3.2).

3.1 Software development method

Software development is not only writing code. Software development is a complicated process comprising many stages, according to Ur Rehman & Paul (2003). To order these stages in a way that it becomes well-organized it is almost necessary to use a method. In the information technology branch there are many different software development methods. In this paragraph a method for this project will be chosen and it will be accompanied with argumentation.

For this specific project I choose "user-centered design" as specified in ISO 13407. User-centered design is often shortened as UCD. As the name already suggest the model has as principle that the user is centered. One of the first that came up with a user oriented approach to develop software were Greenbaum and Kyng (1991). Their method put the user and its knowledge in the center of the development process. Already in the 1990 the percentage of code devoted tot the interface for software products averaged over 50% (MacIntyre *et al.*, 1990). Nowadays Graphical User Interfaces are becoming more important, command lines are rarely seen and the user interacts mostly with the computer by graphics. "To the users, the interface IS the system" (Hix *et al.* 1993).

User-centered design involves the user already from the very first stage (until the end) of the development process and puts designers and users on an equal foot. Througout the stages the designers will learn to understand the users' needs. IBM says about this:
"Throughout the entire development process and beyond, users play a critical role in the design of easy-to-use products. After all, who knows more about which products are easy to use than the people who use them?" (IBM, 2006).

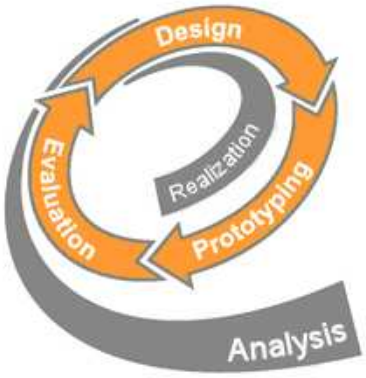
Norman said this about UCD (Norman, 1986 p. 61): "*But user-centered design emphasizes that the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system.*"

The target in user-centered design is to ensure that the product is useful, usable, and meaningful to the end-user. Therefore primary focii for user-centered design are: usability, usefulness, desirability, legibility, learnability, etc. (Merholz and Shedroff, 2002).

User-centered design is an approach with many varieties. Next to that it is an evolving method. There are still companies that (try to) discover new approaches to UCD based on studies or practical experience. Some of the varieties are: Usability engineering (Nielsen, 1995), Human-centered design (ISO 13407), Goal-directed design (Cooper, 1999), Usage-centered design (Constantine & Lockwood, 1999), Contextual design (Wixon *et al.*, 1990), Customer-centered design (Beyer & Holtzblatt, 1998), Cooperative design (Greenbaum & Kyng, 1991), Participatory design (Muller *et al.* 1997).

The different varieties have at least one thing in common; they have different stages in the method. These stages also differ. Many sources (e.g. Rauch et al. 1996) use different stages but generally UCD works like described in picture 1 (next page).

The first stage is the analysis. In this part there will be a meeting with the users, to interview them. The outcome of this part can consist of: user profiles, task flows, personas and/or scenarios. The second stage is the design. In this part of the project one will design the look and feel, including a navigational plan for the application. The users are involved by commenting the design. The design results in a prototype which has to be made. Also the prototype will be tested by users. The results of those user tests will be evaluated and the design has to be changed, until the user is satisfied with the results (realization).



Picture 2: different stages in UCD

Reasons for choosing for UCD are:

- The user is the spill in the method. The application that I will develop will be only used by them. Therefore it is important to involve users in the design process;
- The user-requirements must be found during the project and don't excist already (like in some other models);
- It is the Scandinavian tradition towards design (according to en.wikipedia.org);
- I expect an cooperative attitude from the users;
- There is much documentation on UCD which I can use;
- This whole method is new to me, and it is an excitement to work with it.

There is also a disadvantage of this method. The method it self is not strict, you can use it as you want and the different stage can be filled in according to your own wishes. This however can also be an advantage, it makes the method flexible. Another disadvantage is according to Merholz (2002) that UCD can lead to an application that is most usefull but untenable in a business perspective. This disadvantage is limited in my case. The program has to fit the readers most. Of course it would be very nice if my program is used in the e-paper later on but it's not a disaster for anyone if this will not happen.

There were a few other methods that were investigated in the scope of this project, before concluding that UCD was the most effective for this project. First of all, the waterfall model. The waterfall model is an old model (dating back to 1970) but well known among IT staff. Every stage in the waterfall model follows chronological after each other, without the opportunity to go back. This model however is inflexible and that makes it difficult to respond to changing customer requirements (Sommerville, 2001). For this project the user-requirements are not yet defined and iteration is necessary.
The second model that was investigated was DSDM, far more iterative then the waterfall model. In this model first a feasibility and business study are done, then the user requirements are gathered (functional model iteration), then the design and build iteration in which the product is being coded. After that follows the implementation. After every stage there is a possibility to go back to the previous stage. The problem with this methode is that it more or less expects that there are fixed user requirements. The user is not involved as much as with UCD, which I think is necessary in this crossword puzzle application.
At last I did a short research on the "star-model" as described by Preece *et al.* (1994, pp. 380-381). In this model, designers can start from any point; so by prototyping or by conceptual design, etc. It is an alternative for the analytic (top-down) and synthetic (bottom-up) design steps. I think this model doesn't give me the structure that is needed in this project.

3.2 Tools

In every project tools are needed. In bigger projects they are more often called "resources" and then placed in a wider scope meaning for example money or workhours. Tools, as it is used in this paragraph, are meant as things that you use to complete the project. They are "touchable". In this paragraph the tools that will be used in the project will be named.

One of the most important tools is the programming language. There are many demands for this project when it comes to the programming language. It has to function on an e-reader,

next to that it would be nice if it was also working on a normal PC. In order to make a nice "look and feel" it should be able for the programming language to use with visual aspects (making an advanced GUI). The content of the puzzle have to change every day. It can not be a stand alone program. It has to work in relation with the internet, at least to get up dates or maybe a persisent connection.
Because all the newspaper publishers work in XML it can be necessary to use XML. I think all these demands amass eachother in Sun's Java Standard Edition (J2SE). This language can parse XML. You can size the programs so it is possible to design it for a 8 inch screen. There are two problems with this language. First it is not sure if the application can ever run on the e-reader (and give back input to the application) and second the Java application can not be embedded in XML code but instead the XML will be parsed by the Java application, which is a major difference in approach. The Java application will have to be (down)loaded to the e-reader prior to use. The question if it can be embedded in the e-paper remains unanswered because we've not such a clear view on the programminglanguage or other capabilities of the e-paper device yet.
Next to the programming language there are some more things compulsory to end this project in a proper way. I will give the complete view of what I think is necessary:

- Literature, this is necessary to see what others wrote about subject and to learn the programming language;
- Time to interview users, especially their time can be a problem;
- Some cooperative users (which also speak English or Dutch);
- A development Environment to program and test the Java.
- Many crosswords from Swedish Newspapers to get an idea how a crossword is build up and to fulfill tests the way users use the crossword now;
- To conduct the tests:
    - E-reader to actually run the application (if not available it will be made on a desktop PC);
    - E-mail to communicate with the users;
- Internet, this is necessary for searching information on specific topics.

The solution I chose to realize the program is to build a Java program on a e-reader (or if the reader is not available the tablet PC).
Positive points on this approach are:

- Java is highly interactive;
- Java has the opportunity to design a good GUI;
- The e-reader is portable.

Negative points are:

- Can the program be used on the e-reader?
- Users have to learn a new device.

Alternatives to use in this project can consist of for example a PDA or mobile device as the device. This is however not chosen for because this is outside the scope of the DigiNews project, next to that it is hard to design such a small user interface. Not only the device can be changed but also the programminglanguage can be changed to give the user a better product. For example flash can deliver better graphics and so a better looking product. This however gives some problems when the crossword puzzle has to be saved to fill in later and to receive the "daily update". Concluding, the chosen way of realisating this program is the best for the users. It can deliver the techniques wanted (the high interactivity and many possibilities for younger people and the accesability for the older ones).

4. Design

In this chapter the design of the crossword puzzle will be written down. Desing in this context means the underlying thoughts for the product. In §4.1 users & environment are given. In §4.2 the user task are analysed and presented in a task flow diagram. Paragraph 4.3 describes imaginary users (personas), and in §4.4 scenarios are made.

4.1 User & environment identification

In this paragraph the target audience, with their workenvironment, will be identifcated. In this paragraph the reader will get a view on the current and potential users of the product. It is crucial to identify the user as good as possible because it is the key to everything that follows later in the project. The target audience will be described, including, among other things, experience level, estimated frequenty of use of the product, motives for using the product.

Right now there are no real users of an e-paper. It is still under development at Philips. It is not yet a commercial product. The target group for the e-paper as used within the media-IT research group is very wide. The e-newspaper is supposed to replace the newspapers printed on paper in the future. The group "newspaper readers" is a group that is very divergent. For this project I assume that the same group that is reading the newspaper is also making the crossword puzzles (now and a while). If we take the group as total, the age from the users would approximately be between 15 and 85 years. Also their educational level differs from the Swedish "gymnasiet" to academic levels. To avoid the problem of giving a too general view on the user the total group of newspaper readers will be devided in to three groups. The descriptions are based on my own thoughts and experience. The categories are shown below:

Category A

This category contains young people. The age of those people is between 15 and 30. Many of the people in this category still go in school or have just started their working life. The difference in education is pretty big. This can go from "grundskolan" (for people that are in gymnasiet now) to an academic degree. People in this category grew up with computers. The computer is a part of their daily life and their experience differs but in contrast with the other categories the experience is much. They are young and are very adaptable when it comes to changes. This category is not really the crossword generation and the use of my program would on the average be limited to once a month. Motive for using the program are mostly "pastime". Expectations are high interactivity and a goodlooking user interface.

Category B

This group contains users in the middle age. The age differs between 30 and 65. Most of this people are working. The difference in education and working level are as big as in category A. However people in category B did not (or not many of them) grow up with the computer it is part of their daily life. There computer experience maybe still pretty high due to the practice at work but this does not mean that they learn new things easily. For many people in this category the use of a new technology (such as a crossword puzzle on an e-reader device) is hard to learn. Therefore it is necessary to use the correct metaphors and a good GUI. Their motive is the same as in category A, but their frequency of use is higher, about 1 time a week. This people expect to have the same usability as in a real crossword.

Category C

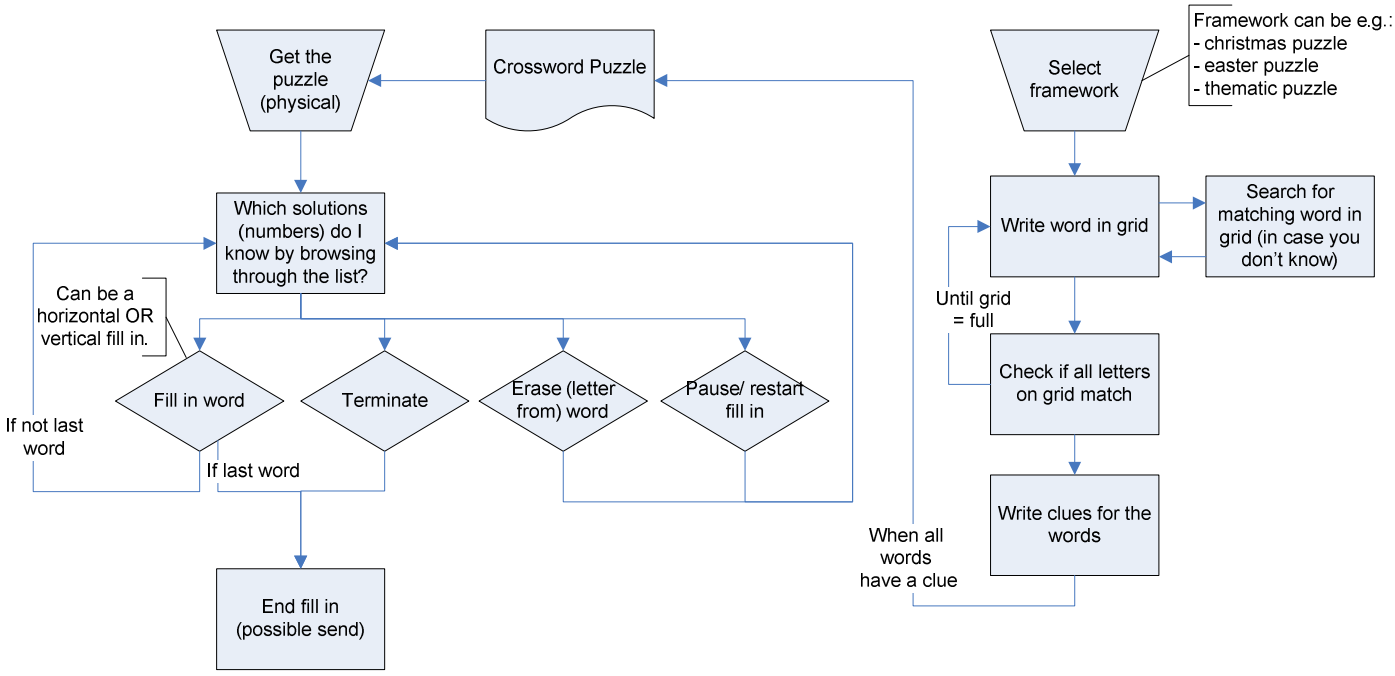
This group contains the older people. Their age is 65+. People in this category are pensioners. Their education and background differs as much as in A&B. They didn't grow up with the computer and most of the people in this category do not use a computer. They don't see the added value on using a PC. They have a mainly negative attitude against technology and changes. Most of them fill in the crossword daily now but if they will continue to do that in the digital way is a big question. Their motives for doing so could be "fun" or "want to learn to have a new experience".

This people need to be helped by providing big letters (sigth is less good) and a simple interface. Their expectations are to only fill in the crossword puzzle (and return the answer with the mailman.

The users' environment is the same for all the categories. The users differ much in the place where they use the crossword puzzles. Some people make them at home, some make them in train, etc. This points out that the puzzles have to be portable, with the e-reader as device this shouldn't be any problem. Next to that the users can save them and open them another time to complete the crossword. The environment can be noisy, so sound within the crossword puzzle is not an option.

4.2 Task flows

In this subparagraph the task of the user will be analysed. The task the user completes in general is making the crossword puzzle. In this task there are several subtasks that need to be intentified. On one hand this information is necessary to know to make a good program, on the other hand it is important because the developer then get a good knowledge of the user and insides on the task at hand. This makes clear why I think the task flow is so important for this project. From the task analyses this scheme below is destilled:



4.3 Personas

Personas are imaginary people. The developer uses this description to feel along with the user of the product. It is like the developer can see things through their eyes. For projects in which the user requirements are not quite clear (like this project) it is easy to make personas. The personas help the developer to see through the eyes of potential users. I think that personas can help me to get other views on topics that are important for the design of the application. The personas are also helpful when there is no real user available when making a design descision; therefore I chose to use the persona technique.

For this project three personas will be used, one persona for each category of users (see §4.1). In this paragraph you find those three personas.

Nathalie Darcy:

Nathalie is 19 years old. She is born in Hampshire, England. She lived in England for 12 years and then she moved to Sweden with her parents. In Sweden she studies Psychology or as the sweeds call it "psykologi" in the University. This is her second year of study. She is still living with her parents (that live close to her school) in Umeå. Next to her parents she has two smaller brothers (13 and 8 years old). She uses the computer in school to make reports and to search for

information on the internet; school also has some educational programs that she uses. At home she uses the computer mostly to communicate with her friends and other relatives. MSN is in her favorit list, together with hotmail and www.lunarstorm.se. When she is in school she makes a crossword puzzle sometimes. There is always one in aftondbladet in schools library.
She loves to see her friends, go out with them and make much fun. Normally she can do that in Umeå but sometimes they travel to Stockholm to party a weekend long. One time a year she travels to an international music festival (like Roskilde in Denmark or Rock Werchter in Belgium). In order to do this she need money, to obtain this money she works in weekend in the hospital, she's cleaning the dishes there. It is pretty fun with the young collegues there. Her favorit quote is: "The biggest adventure you can ever take is to live the life of your dreams."

Anton van Hoensbroeck:
Anton van Hoensbroeck is 37. He is born in Rotterdam the Netherlands and still live there. He's very fond and proud of his city. He and his wife (Silvia) live in a neighbourhood called "Feyenoord". They have 2 kids together, called Sjaak and Marije. He likes his kids much and wants to spend much time together with them. In weekend Anton plays football with Sparta (senior team). He thinks it's good to move some and playing football is just to relax. One time in the week in eve he's also doing line dancing with his wife. He always has to keep this day free because the rest of the week he is really busy with his job.

Antons works for Etol Nederland. They sell soap for industrial purposes. Anton has many clients like minsteries, hospitals, catering services, etc. He is traveling from one client to the other the whole day. His working area is mostly the middle of the Netherlands. Sometimes he has to go on meetings to Germany because Etol is a german company (with a department in the Netherlands). The car Anton has (a BMW 525d touring) looks like a complete office. In the back he has the try-out soap packets for promotion and the first-aid kit to repair damaged soap systems. In the front he has a laptop, a mobile phone, a navigation system, a PDA, etc., everything to communicate with collegues or clients while driving the car. Before he goes to clients he always drive his car to the company building in the morning to synchronize his PDA and to read some company information. In his lunchbreak Anton reads the paper. When he has nothing else to do at a parking place he used to make the crossword puzzle from his morning paper (de telegraaf). Sometimes he plays a game on his laptop in the evenings or in weekend. He uses the computer much for different purposes. Still at home he's not the best in it. His son Sjaak is studying ICT and knows much more about it. Anton didn't study Informatics; he followed a study in marketing. He got his Bachelor degree when he was 22. His favorit quote is: "I've learned that you can't have everything and do everything at the same time."

Sven Carlsson:
Sven is 67 years old. Sven is a real sweed. He lived his whole life in Genarp, close to Malmö. He has 6 childeren but they all left the house. His children are between 40 and 23 years old. Now he is living together with his wife. Sven had one older brother but he unfortunately died when he was 74. He has one sister left but she lives in Åre so they don't see eachother so much. Sven's children live in Kalmar, Malmö and Lund and they visit him regularly. Sven and his wife moved for some years ago, when all children moved out of the house. Now they live in a small house, but they love the house and the surrounding. Before Sven was a farmer. He used to grow vegatables and other food, differing from potatoes till carrots. Sven stopped when they moved out. Now he's enjoyning his retirement and has a small garden because he can't really stop, being a farmer is in his blood. Sven don't have an education. In his early days his parents didn't have the money to send him in school. There were 12 kids and not all of them could study. His parents raised Sven and thought him how to grow food. He learned most of his father, who was a farmer too. Now one of his sons has taken over their farm, it is a real family business.

Sven never uses a computer. He thinks it's hard to understand does machines, and on his age it is hard to learn new things. His son is introducing computers in the farm now and if Sven sees what you can do with them he thinks it's "wonderful". Now Sven has so much time left he reads the newspaper carefully, he's especially interested in wheater forcast and local news.

He also likes to make the crossword puzzle. Sven is a member of an athlectics club. On his age he's not doing it him self of course but he is in the board of the club and he is involved in making the club's magazine. He used to design crosswordpuzzles for that purpose. It takes him some hours to design one but it is really appreciated by all the memebers. Sven works all his work out on paper and someone else digitalize it. Svens favorit quotes are: "All programmers are playwrights and all computers are lousy actors." And "Computers are useless. They can only give you answers."

4.4 Scenarios

A scenario is again something imaginary. It describes a case in which the user would use the program. The developer tries to look in the future and visions the user using the new software product. Cooper (1999) describes it as: "Daily Use" scenarios that outline the actions a user will perform. In this way the developer is forced to think like the user. Cooper (1999, p. 123) says about this: "We forget our own education, ability, training, and tools, and imagine ourselves as having *his* background instead".
In software development it is easy for the developer to overlook things that are crucial to the user. The developer is mostly not well known with the task the application is being designed for. In this case the scenarios are helpful. It can give an insight on how users will use the program, it can als be used for testing purposes to see if the developer didn't overlook something the user need to have. For this project it is quite important to see or vision how the user will use the program. I choose the scenario technique because it is close to "natural language". It is easy for everyone to understand and it is easy to use for testing purposes. It is a straight forward technique that can be extremely usefull when designing and testing the application.

For this project I created two scenarios, where two of the three personas are used. To underline the difference in use I took the personas from Anton and Sven.

Hit the road
It is a great Wednesday morning when Anton parks his car next to the bridgerestaurant on the A4 to Amsterdam. He's working today and it is just 08.00 on his watch. The sun is coming up. He expected that he would have some traffic jams so he went up early totally forgetting his coffee. He is expected to be at the client at 09.30 so he has 45 mins left to have some coffee. He walks in, ask for a black coffee and get it. He sits at the table watching the traffic. He opens his suitcase and suddenly he remembers that he didn't even read the paper this morning. It's great that everything is digital nowadays. Of course the bridgerestaurant is equipped with a wireless network. After reading todays news he still has 20 mins left. It's time to do a little crossword puzzle he thinks. First he takes a new cup of coffee. The program opens itself and Anton knows already some answers by browsing the list. He fills in the correct answers. Now he has to think. That is hard on this morning. Know he already filled in some words it's easier to find the others. The great thing about this new program is that you can easily correct your mistakes. Totally absorbed by the puzzle he is shocked when he looks at his watch; almost 09.10. He closes the application by saving the puzzle. I will make this at home later he promise himself. Now he has to hurry to the customer. When he comes home in eve he opens the program again. He HAVE to finish this puzzle. He is too tired to give correct answers but his wife is there to help him. It is much more fun with both of them. They fill in the puzzle. Anton feels done now. He sends in the puzzle to win the prise and put off the program. Maybe tomorrow again when there is a new puzzle.

Behind the geraniums
Sven is having a normal day today. He got up at 09.00 this morning to look after his plants. He is doing this every morning. He thinks if you treat them good that they will grow better. This morning his plants showed much progression in their growing. He is really happy about it. When he comes home he tells his wife when they are having coffee together. The clock points 10.30 then. Together they make the shopping list because there is not so much in the refrigerator anymore and in eve their son will come and visit them. They prepare the dinner for it. After the coffee his wife is going to shop and Sven does the dishes. After he's finished he wants to relax a little. He sits down in their big arm chair. Sven likes this chair much. His eyes fall on the device next to him. It is the e-reader. He got one from his son a few weeks

ago. After the free training for older people he knows how to handle it. He's not really good yet but he thinks it's interesting to learn it a little more. Sometimes he calls his son when he has questions about it. He starts the e-reader and in the mean time he takes on glasses. Reading without them is really hard for Sven. He makes the day puzzle. It is easy for someone like Sven. He makes the puzzles for so many years now. He's finished in 15 minutes. He knows that he have to make a crossword puzzle this week for the club magazine. He decides to do it now. It is a big challenge for him but he will try to do it on the e-reader. He starts by taking the correct frame for the puzzle. Then he fills in the words and descriptions. Silently he says to himself "design was never this easy". Just before his wife comes home he is finished with the puzzle. Now the only thing he has to do is send it to the right person so it can be with in the next magazine. With the e-reader and this application that is no problem. Sven sends the puzzle and he's proud of himself.

4.5 Prototyping

A prototype is a model. In the case of software development it is a model of the program. A prototype is made because it is cheaper then building a complete program. The prototype can be used as tool to communicate design with (potential) users. There are different types of prototypes; there are low-fidelity prototypes and high-fidelity prototypes (Preece *et al.* 2002).
Low-fidelity prototypes are cheap and easy to make. This can be for example a scratch. One can communicate the design with the users. The users then get an idea of the possibilities there are. They can comment the prototype or explain why they prefer one design above another. High-fidelity prototypes are prototypes that look very much like the final product. They are made to show the user what they get (Sommerville, 2001), to make small changes or for testing purposes. If the software development process is followed in the right way it shouldn't be necessary to make many or a big change before the high-fidelity prototype becomes the final software product.
I used both low- and high-fidelity prototypes. It is a good way to communicate design and to find out what users think and what functional requirements they have towards the final product. It cost less time to make a sketch then make a program and the user feedback from a sketch can be very useful.

4.5.1 Low-fidelity

First I developed 4 low-fidelity prototypes. These were sketches of possible solutions to design the application. I draw every low-fidelity prototype on a half of an A4 paper. When designing the prototype I read a persona first and then I imagined that I was this person. Then I took a scenario. The scenario is a possible situation in which the program is used. In that situation you feel like you become the user and know how you'll use the program. Then I reviewed the task flow. This gives insight in the tasks the user will complete, and in which order. On the moment I thought I had all information complete I started to draw.
When drawing I kept an eye on the 5 usability attributes as described by Nielsen (1995). Equally important to the usability attributes was the theory about mental models as described by Preece *et al.* (1994). They state that a mental model is a model people have of themselves, other, the environment, and the things with which they interact. People form those models through experience, training and instruction. In this project the mental models are important because the user will see a new type of software. The user relies however on earlier experiences with the crossword (for example in the newspaper) and the use of software. The user will expect specific things to happen by running their mental model.

In the first prototype (see all the prototypes in appendix 1) I placed a set of buttons on top of the screen. There was a large field (the actual puzzle) to the left and the descriptions of the words were placed on the right side of the screen inside a scrollable area. The buttons were located on top because they attracted the attention and for users that are quite used to the computer (such as two of the personas) this is a logical place. In e.g. Microsoft Word, Internet Explorer, etc. the buttons are placed there too. This should be in accordance with the mental models the user already has from earlier experiences and it improves the usability attribute: memorability. The descriptions of words were placed in scrollable areas to fit in a pretty small screen.

In prototype II I changed the place of the buttons. This was done to see if older people were more comfortable with that. Another place of the buttons could improve the satisfaction (also an usability attribute) because it's a more logical order to work from above to below. At first sight it is makes a more calm impression (also improving the satisfaction), but more experienced users probably have to search for the buttons.

In prototype III the buttons were exchanged for a menu. This menu had the same possibilities as the buttons. The words and the puzzle did not change place. The menus are both easy for experienced users and new users. For old people (like one persona) they're maybe hard to see (because it is smaller and written text).

In prototype IV the buttons were back but had another place. They were placed on the left side of the interface under each other. The puzzlefield was located to the right next to them and the descriptions of the words were under the puzzle field, next to each other. The descriptions of the words were placed below the puzzle because this happens in some puzzles as you can find them in the newspapers. The people that are used to puzzles on paper would maybe appreciate this (two personas). Again, in appendix 1 all the prototypes are given.

The prototypes were kept as easy as possible to make it suitable and understandable for everyone. A prototype is only to show design options and possibilities. Moreover the efficiency and the memorability can improve when having a simple application. For some personas in some scenarios, a simple GUI can improve the satisfaction. Older people for example want an easy to use and clear GUI. Not to much extras or technological gadgets. The application will have much better results if kept simple.

Evaluation

For the evaluation I used the "quick and dirty" evaluation technique as named by Preece *et al.* (2002). With this method designers informally get feedback from users. In the "quick and dirty" evaluations the emphasis is on fast input, rather than carefully documented findings. Preece *et al.* (2002, p. 341) state that "*getting this kind of feedback is an essential ingredient of successful design*".

To see what possible colors the users like, I also combined some color schemes. In Appendix 1 you can see the prototypes and the colors. After I made the prototypes I asked 4 people to look at them, the exact test method and procedure is described in chapter 5. Two were in category A (between 15 and 30 years of age), two were in category B (30-65 years old). On average they all had the same computer experience. 3 of 4 subjects were inhabitants of the Netherlands and conversated with me in Dutch, one however was Swedish and she converstated in English. From this small test some interesting conclusions could be drawn. There was a difference in the wishes of young and older people. Both the two young people chose independtly from each other for prototype 1 with the blue color scheme. In the argumentation they say that it is easy to have the buttons up on the screen and that scrolling for the descriptions of the words is necessary. One commented that he took prototype 1 because the lay-out was easy.
Younger people choose for blue colors. According to Color Wheel Pro (2006), "*blue is the color of the sky and sea. It is often associated with depth and stability. It symbolizes trust, loyalty, wisdom, confidence, intelligence, faith, truth, and heaven"*. For the newspaper this seems to be the words that should be close. Trust and wisdom are important factors. One of the respondants says that the contrast between the green and blue was very good.

The older people chose different one preferred prototype 3 and one prototype 4. Prototype 3 was, according to one person much easier. She seems to prefer a roll-down menu above buttons on the screen. Her statement furthermore state that the Graphical User Interface have to look easy. She seem to want as less as possible on the screen. Prototype 4 was also chosen by one person. The argumentation for chosing this one consisted of the statement that a user reads from left to right. The descriptions of the words are the first you want to see (and should, so, be positioned on the left hand in the GUI). In prototype 4 the GUI provides a possibility to see the descriptions first.

In contrast to the younger people one middle age person chose for the red color. According to Color Wheel Pro (2006) *"red is the color of fire and blood, so it is associated with energy, war, danger, strength, power, determination as well as passion, desire, and love"*. For the crossword puzzle energy and passion could be applicable. Another person in the same category however chose the blue scheme, like the young people.

The opinions about the prototype are very scattered, young people seem to have a slight preference for the first prototype. Because the users not only provided me with their choice but also with their arguments the good things from each prototype can be derived. Remarkble enough most of the arguments are not in contradiction to each other. From the user comments a first concept of user requirements can be drawn up. Users state that they don't want many components on a screen (easy to learn and remember, ease for the eye). They also state that an interface is read from left to right, what should be seen first should be placed left (fast speed of performance). The colors should be cool instead of warm. The last but not less important requirement that can be learned from the comments in this session is that the program must fit the users existing mental model (earlier experiences with puzzles and computer programs). This results in the following list with user requirements:

UR1. Don't place to many components on the screen;
UR2. What should been seen first should be placed left;
UR3. Use cool colours instead of warm colours;
UR4. Built the application on the users' existing mental model.

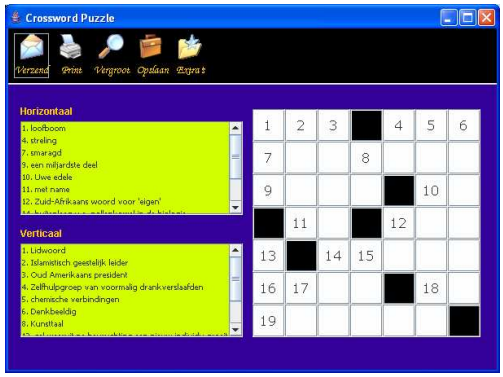
By using the low-fidelity prototype and the quick and dirty evaluation the users were not able to try the interactivity no one however stated that they missed a function. I checked the functionality by using the task flows. In the task flows one can see the path the user uses when completing a task.

The users indicated that buttons are more usefull then a menu and the buttons are preffered to be positioned on the upper side of the screen. The descriptions of the words should be placed on the left hand and should be scrollable. The size of the puzzle field was good and the puzzle field should be located on the right side. With this input from the low fidelity prototypes were converted into a first version of the high fidelity prototype. Preece, *et al.* (2002) describes a high fidelity prototype as a prototype that looks like the final thing. In this case I'm designing software and so the prototype is ought to be a program. It is according to Preece *et al.* 2002 completely functionally and fully interactive. This program can then be again tested by users and one or more iteration(s) can convert the high-fidelity prototype into a final product.

4.5.2 High-fidelity

After the low-fidelity prototype was built and the user comment on those prototypes was collected it was time to build a high-fidelity prototype. The high-fidelity prototype was build with the programminglanguage JAVA. Java is platform independent and easily exchangeable between systems. There is much information about programming in Java available on internet.

I took low-fidelity prototype I as the basis to start designing with. This was the prototype that the users seem to like most (according to their own comments). I built the graphical user interface exactly like it was in that prototype (see appendix 2 for the high-fidelity prototype). Then I took the user comments on the low-fidelity prototypes and started to change. The first buttons were the first thing I changed. I made a toolbar with icons. This was to make the user interface easier to learn and remember (one of Nielsen's usability attributes). Visual icons can always be remembered and found easier and faster then text when it comes to buttons.



Picture 3, high fidelity prototype.

The images I used were an envelope for the send function, a printer for the print function, a magnifier for the enlarge function, a briefcase for saving and loading a puzzle and a map with a star for extra's. The next thing I changed was the position of the descriptions of the words. According to the users you read from left to right. The descriptions need to be seen first so they should be placed left. I did this to improve user satisfaction and to fit the mental model from the user (which should match a physical crossword puzzle). The users clearly state that they wanted cold colors. I sticked to the color scheme's I presented the user first and chose for the same scheme as most of the users.

After implementing the graphical user interface I added the interaction. Here I used the task flows (see §4.2). In the taskflows one can see that after the puzzle is made visible the user will browse the descriptions those the user knows he will fill in. I chose to start with filling in the descriptions and make sure that the list is scrollable. After doing that the user can do many things, fill in a word, terminate, erase a letter or pause the crossword puzzle.
I chose to build up the puzzle field first, so the user can fill in words. Correct and usefull interaction in this piece is necessary. Nielsen (1995) talks about effiecieny, it should be easy for the user to complete the task. I made the descriptions clickable. If one clicks on a description they light up in the puzzle, so the user knows where to fill in the letters. If the user want to fill in a word the navigation is important. I made the crossword puzzle in a way that if the user clicks a word and then fills in a letter the next field is selected. This makes sure that the users don't have to click everytime after filling in a letter.

After that I chose to implement the terminate function. The user can terminate the program according to their mental model by clicking the cross right up on the screen (like in Windows applications). Nielsen (1995) writes about memorability and learnability as being usability attributes. When a function (or like here the cross to close the application) is close to the users' already excisting knowledge it is easier to remember for the user. After the closing button also the erase letter function was implemented still according to the task flow, if the user clicks in a field he can erase a letter by using the keyboard. The pause and restart function was also implemented. I used the personas to think about this function. If you read the persona and the scenario from Anton you can clearly read that he don't always have time to finish the crosswords. If the designer thinks from this point then it would be easy if you can save the crossword. That's also the way I implemented it. The user in the high-fidelity prototype is able to save the crossword puzzle after filling in words. When the user starts the program the next time an easy click on the briefcase makes it possible to load a saved puzzle. The last thing the taskflow showed was a possibility to send the crosswordpuzzle. I implemented this function. The user has to give his own e-mail address in a new window that pop-ups and close automatically, than an e-mail is send to the puzzle designer. The design part from the task flow could be fitted under extras but isn't implemented in the high-fidelity prototype because I received to less input on that. I can't find users that actually build puzzles. Furthermore it would take much time to implement this and most of the users won't make use of it. I decided to focus on the functions that do matter to the users that fill in the crossword.

There were also some functions that couldn't be found in the task flow. Mostly because the task flow was an analyse of how things work now and not in the digital world. The digital world creates new possibilities, which can be used. I read the persona from Sven again and also reviewed the scenario "behind the geraniums". Then I came to the conclusion that the program should have an enlarge function, which makes the user interface better readable for older people. Such functions make it easier to use for special groups of people and that is an improvement for Nielsen's usability attribute satisfaction (Nielsen, 1995). I implemented this function in the high-fidelity prototype. It makes both the descriptions and the letters in the puzzle field bigger. Some users still want to be able to fill in the crossword on paper. This function can be found in the "print" function. In the high-fidelity prototype the puzzle field is printed (with the already filled in words).

When all the functions were ready I made sure that errors were catched. This is in accordance with Nielsen's usability attribute "errors". A designer should try to prevent errors but if they occur they've to give the user a meaningfull message. Errors were catched for problems with

the send function (when the puzzle can not be sent) and with the load function. If the user never saved a puzzle he can't load it.

As finishing touch and to make it easier for the user an install program was used to pack the application. This install application gives a nice visual presentation and installs the files on the users' harddisk.

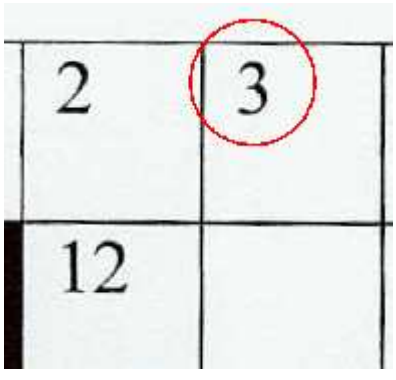
Evaluation

When the high-fidelity prototype was finished and packed into a nice installation program it was time to test it. This test was done to see if the prototype fit the users' needs. The high-fidelity prototype was the last thing the user could comment before the final product would be released. The same test users as for the low-fidelity prototype were used. This was done to create stabile comments and so they could see the effect of their comments. These were in total 4 subjects (2 in category A, and 2 in category B). In addition to that I asked 2 more testpersons, one in category A (between 15 and 30 years of age) and one in category C (65 years of age and older) to also test the high-fidelity prototype. The first subject was a student with a high computer experience. The older lady had not that much computer experience, but had an experience above the average in this category. Again all of the people conversated with me in Dutch, except one. She was Swedish and therefore conversated with me in English. The exact procedure about the tests is described in chapter 5.

In the evaluation of the low-fidelity prototypes there was a difference between the older and younger people. There was also a divergence between the opinions inside those groups. In this evaluation this differences were marginal. The general opinion and the comments I received were generally the same throughout the different categories.

The use of color in the application was decided after the choice of the users in the evaluation for the low-fidelity prototype. It should be cold colours. In this evaluation there was noone that objected to the use of cold colours. The user group that also commented the low-fidelity prototype thought that the color was well used and according to their whishes. There was one of the "new" testpersons that said that the color of the buttons (black) didn't match with the background (blue/purple). The cold color set was good, but he wanted more contrast. Two users commented that the lettertype that was used for the buttons was unclear. The description under the buttons was very useful, but in this case it was hardly readable. All the test persons agreed that this application had everything that they could do with a real crossword puzzle. The users were asked if they missed a function, but none of them said they did. In this I conclude that the functional requirements for this project are met.

What the users were not able to comment in the low-fidelity prototype was interaction. This was the point where I received most comments in the evaluation of the high-fidelity prototype. Four out of six testpersons commented that it is possible to erase the numbers and that when you filled in the first word that the numbers disappear. The numbers are very important for the users because the numbers refer to the descriptions. It is very uneasy for the user to miss this numbers (which decreases the satisfaction, a usability attribute from Nielsen, 1995). The miss of numbers also points out that it is not clear for all the users that they can click on the descriptions of the words. The place of the words can namely be lightened up by clicking on a description, so that the user knows where to fill in a word in the puzzle by the description given. One user said that it wasn't clear in the beginning that the descriptions were clickable, but when she discovered that it was possible she judged this as very useful. She came up with the suggestion that the puzzle field should be locked so that one has to click on the descriptions. Other critics to the interaction were that the print function is usefull but that it is not possible to print the description of the words. This makes it impossible to fill in without the computer (the reason for printing it out). One user also commented that it would be very nice if the letters that were filled in by the user would



Picture 4: Important number

be shown in capitals directly, the user holded the SHIFT button for every letter. Another user had the last comment to the interaction. He stated that it was not clear how the save/load function worked. It was unclear when you saved a puzzle or when you were able to load it.

When I asked the users if they would use this application in a real situation they were scattered. Two people said that making puzzles wasn't their favorite thing to do and that they wouldn't use it. The other four said that they would use an application like this but added the condition that it should be possible to receive a daily update to the puzzle.

From this user evaluation one can draw the conclusion that there were fewer comments regarding the interface but that the interaction is demanding some serious changes. This is however not very surprisingly because the user didn't have the possibility to see or comment this before. From this test I want to draw some general conclusions that are shared among all of at least most of the testers. As far as concerning the GUI it must be said that readable lettertypes must be used, especially when designing for an older category of people. Towards interaction there are three things to say. First it is important that things that are crucial for the effiency of the application do not disappear from the screen and there shouldn't be a possibility to erase them either. The second is that it is not automatically clear to a user that items in a list are clickable. This must be made clear. The third and last is that one button shouldn't have 2 functions. It is better to have two buttons in that case. This result in the following user requirements:

UR5. Use well readable lettertypes;
UR6. Things that are crucial for the application should not disappear from the screen and it should not be possible to erase them either.
UR7. State that items in a list are clickable;
UR8. Do not make one button for two functions. Use two different buttons instead.

With this new guidelines towards an application that completely meets the users' whishes and demands the last convertion will be made. The high-fidelity prototype will be changed in the final product. The user gave comments on the GUI and on the interaction. All the users comments will be read over again, the user requirements that were found will be used and all changes that are technically possible will be incorporated in the final product.

4.5.3 Final product

In the final product the user comments on the high-fidelity prototype were incorporated. I started with the lettertype in the menubar. This lettertype was according to users hard to read. A more readable lettertype was selected (see picture 5).. The next change was the statement that items were clickable. In the lower part of the GUI a short line of text will be shown when the user start the application so the user knows that he should click on a description. The third and last change was the use of the buttons. The load and save function was splitted into two buttons (see picture 5).

Picture 5: menubar

There was one change that was impossible to implement. The user didn't want the numbers to disappear but instead they wanted them in the right corner of the textfields. In this application the fields are textfields and it is therefore impossible to have one text over another, or in the upper right corner.

The whole technical solution to the final application will be described in chapter 6.

5. Testing method

Within user centered design the user is involved in almost every stage of the development process. This chapter is about the way (possible) users gave their feedback on the designs and how tests were conducted.

Method:
To conduct the user test in this research I chose for Field Study. Field studies are characterized by taking place in the real world (Kjeldskov & Graham, 2003). A field study is a research method where the subject to be tested (in this case the prototypes) is used in a situation where it could be used normally too. In this case, the prototypes are used by users from their own home, behind there own computer. Field studies increase the realism (Kjeldskov & Graham, 2003). In this evaluation test users could take as much time as they need and they were able to do it on a time that fitted them. Disadvantages include limited control of the expirements and complicated data collection.

I chose this method because the increased realism is very important when using user centered design. User centered design is used to make a program that fits the user. In my opinion you can't make a program that fits the users' wishes without a look into the real world of the user, therefore I thougth that field research fits best here. Another reason for chosing Field studies was that the user can test the product thourogly, he or she was not restricted by the researcher's time and the tests were not time dependent. I considered the disadvantages (as mentioned above: limited control and complicated data) of the method and tried to decrease them as much as possible. The limited control was decreased by the size of the application. The crossword application that was developed in this project was relatively small, so the possible comments were less in amount as well.

The data that this method delivers was qualitative data. Qualitative methods give a deeper understanding then purely quantitative data, states Silverman (1999). He continues by saying that words and images are ought to be more valuable then numbers in those methods. The goal for this project (as stated in chapter 1) is to come to user requirements and design guidelines. I therefore think that it is more important to understand what the user is thinking and saying and why they do so rather then discover the difference between numbers or user rates.

The user was free to comment on everything and was asked to have special attention for the use of color in the application, the interaction within the puzzle, if all the functions worked like the user expected them to work, if they missed a function that is possible with a normal crossword puzzle and if they thought that they would use the program in a real situation. This resulted in very rich data. Mostly because of the combination of the freedom the user had to comment everything and also the guide on what they should pay attention to. The data this method results in is very complex and can differ amongst users. If I didn't understand the user I asked on so their statement was clear. In the end I tried to generalize the user statements, so that they can be used in this project knowing that it is a generalized statement and not based on one opinion. The data I got back from the questions as stated before was very rich and impossible to translate into quantitative data without losing the meaning.

Materials:
To complete the different tests, users used their own desktop computer (an e-reader device was not yet available), at home. The first thing that was commented by the users was the low fidelity prototype. This was send to them in a Word-document. The second thing that was commented by the users was the high-fidelity prototype. An EXE installation file was send to them including instructions on how to run the program. These tests took place at different times. E-mail was used to communicate there thoughts and comments with me.

Subjects:
There was a core group of users that provided me with feedback. They were involved from the beginning to the end of the development process. This core group counted only 4 people. This

was because they fitted the profile of the targeted user group and because there voice could really be heard. The core user group consisted of 2 young people (under 30) and 2 people that were between 30 and 65. The young people were both students, from the older people one was a security officer and the other one was unoccupied. All subjects had a fair amount of "computer knowledge" and were familiar with filling in crosswords. All the subjects were living in the Netherlands and therefore communication took place by e-mail, in Dutch.
The high-fidelity prototype was tested by 2 more people who were not belonging to the core user group. One was pensioner and belongs to the group 65+, one was a student and belong to the group 15-30. The same procedure was used as with the core group, but this group never saw the low-fidelity prototypes. There comments were not pre-judged by prototypes they saw earlier.

Procedure:
The first contact with the core user group was when I made the low-fidelity prototypes. The low-fidelity prototypes were scanned in and the color scheme was added. This was put into a Word document and sent to the users e-mail address. The user was asked to chose one prototype and give arguments why they choose just this prototype and the same for the color scheme. The user comments were based on their feeling. *Why* they chose one prototype or color scheme over another was really important in this stage. All the users' comments were collected and were together with the low-fidelity prototypes input for the high-fidelity prototype.

A high-fidelity prototype was designed and coded. An installer was used to pack the program to an EXE file, so the user could easily install it. This EXE-file was sent to the users e-mail address. It was accompanied with a short instruction. This instruction stated that the user had to dubble click on the EXE, then go through the installation process and that the crossword program was then available from the start menu. If the user couldn't open the program correctly a link to Sun's Java installer was given, after installing that package the program runned correctly.
The users were asked to give as much comment as possible. Though they were spefically asked to comment the use of color in the application, the interaction within the puzzle, if all the functions worked like the user expected them to work, if they missed a function that is possible with a normal crossword puzzle and if they thought that they would use the program in a real situation.
When testing the application the user could take as much time as he/she wanted and all subjects were able to test the product thourougly. The users' comments were sent to me by e-mail. A form was not used, the users commented in their reply e-mail under the questions I wrote them. These questions were:

1. What do you think about the use of color in the application?
2. How do you judge the interaction within the puzzle?
3. Do all the functions work like you expect them to work?
4. Did you miss a function that is possible with a normal crossword puzzle?
5. Would you use the program in a real situation?

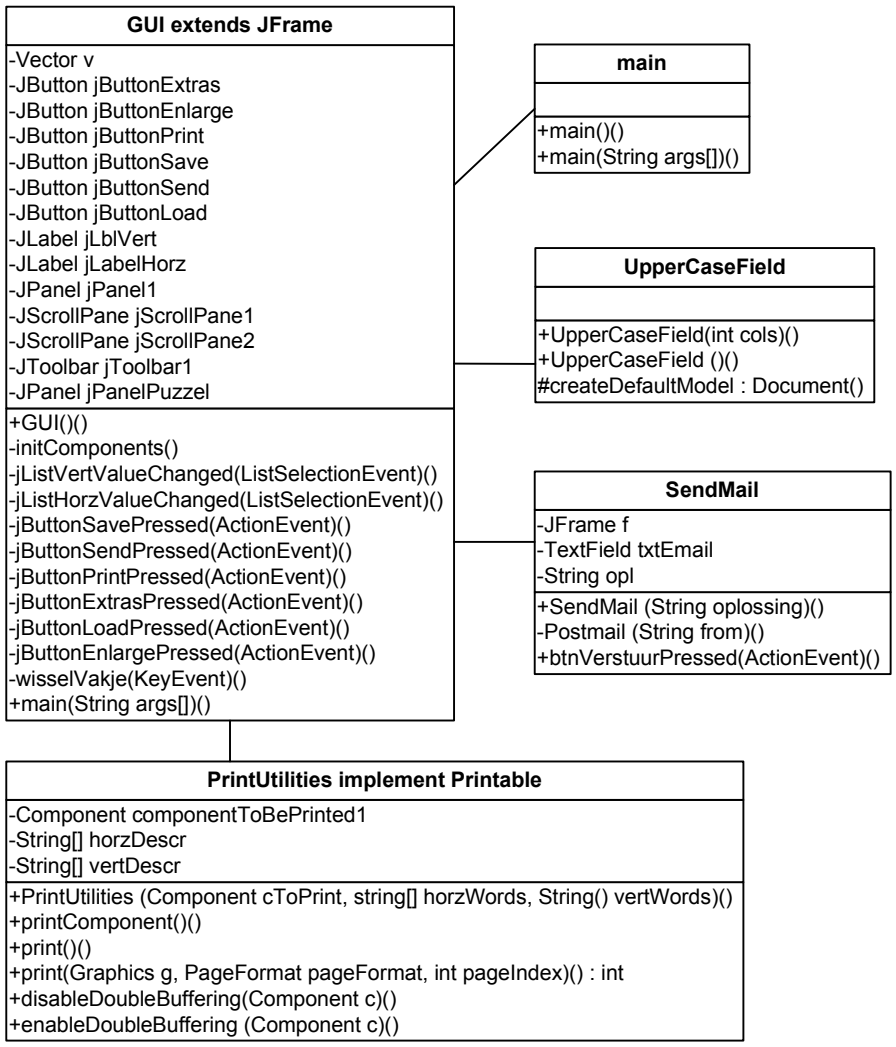
I collected the user comments and used them to change the high-fidelity prototype into the final product. User comments that were unable to implement (because of technical or time limits) were written down in a document that was a part of the final result of this project.

6. Technical solution

In this chapter the results from this project are presented. This chapter is split in two different parts. The first part is a technical description of the program (§6.1) in which the final solution is being spoken of. Parts of the code behind the application are also subject of this paragraph. In the second part there will be spoken of the user requirements (§6.2). The user requirements were found during the project in different stages, in this paragraph they are united and desribed.

6.1 Program

The technical solution is the subject of this paragraph. The full code is to big to show in this documentation. The technical solution is made with JAVA, a programming language both for the web and for stand alone applications. In this case a JAVA application is used (different from a JAVA applet). In the UML scheme that follows below it is possible to see how the program functions.



Picture 4:
UML scheme of the application.

The program start with calling the class main and the function main(String[] args). This functions initiates a new instance of the class GUI (by calling GUI()) and makes the GUI visible. On the GUI the first thing that happens is that initComponents is called, which initiate all the components and put them on the screen. The other functions in the GUI method are listener methods for the buttons, lists and for the fields. These listeners take care of the interaction, for example if a button is pressed.

The puzzle field on the GUI is build up from textfields. These JTextFields should only contain uppercase characters and therefore the class UpperCaseField is used. This class inherits all methods from JTextFields and add own functions to make sure that all characters entered into the field are converted.

The other two classes in this UML scheme are called from the listener methods. If the button send is pressed for example the class SendMail is initiated. The listener passes a string (the solution of the puzzle) to the constructor SendMail(String oplossing). From there the SendMail class will open a new window, in here the user have to give his e-mail adres. This e-mail adres and the solution are used to send the mail to an, in the code, specified e-mail adres.
When the user clicks on the button Print the class PrintUtilities is called. This class prints the puzzle and the words. First the Component jPanelPuzzle is passed to the PrintUtilities class, this is the main component to print. Also two string arrays with the descriptions of the words are passed so that they can be printed too. In the PrintUtilities class the printJob is made, the descriptions of the words are being placed on the paper and eventually the user can print by specifying a printer and clicking print.

6.2 User requirements

User requirements have little connection to the functional requirements. Functional requirements are the functions that an application has to have. The functional requirements state what the program should be able to do. In this project a functional requirement is for example: "fill in a word in the puzzle". User requirements have to do with what the user thinks is important. It has more to do with *how* the user can work with the functional requirements.
In this project the user requirements were not known on forehand. The user requirements had to be found during the development process. The method used (user centered design) was very suitable for this purpose. The user requirements were found by involving the user in the design process. The users saw two different prototypes and were able to comment those prototypes. The user comments were collected and were generalized into the user requirements for this project. The user requirements found during this project were:

UR1. Don't place to many components on the screen;
UR2. What should been seen first should be placed left;
UR3. Use cool colours instead of warm colours;
UR4. Built the application on the users' existing mental model.
UR5. Use well readable lettertypes;
UR6. Things that are crucial for the application should not disappear from the screen and it should not be possible to erase them either.
UR7. State that items in a list are clickable;
UR8. Do not make one button for two functions. Use two different buttons instead.

UR1 t/m UR4 were the result of the comments on the low-fidelity prototype. The user was only able to comment on a paper sketch that was made. This resulted, as you may have noticed in user requirements that were more interface related. UR5 t/m UR8 were found during the evaluation of the high-fidelity prototype. The user now had the possibility to comment both interaction and interface. This resulted in more interaction related user requirements. The results from the low- and high-fidelity prototype are not in contradiction to eachother, which means that this picture gives a stabile base for future use.

The user requirements give a sufficient picture of what the user believes is important. These user requirements can be used as input towards further development of the crossword puzzle for the e-newspaper and possibily even towards for design in general. The user requirements found can be used in addition to other guidelines (such as the usability attributes from Nielsen). Nothing in this research pointed at a conflict between other guidelines and the guidelines derived from this project.

7. Conclusions

In this chapter I will review the whole project. It was a very innovative project in which I learned much and discovered problems as well as having funny moments.

The project on itself went well. The time that was planned was used. The last weeks I had even some time left to update the documentation and change the concept documentation so it could be better. There were almost no parts that have to be dropped out. The only part that was impossible to do was the option that one could design a crossword puzzle that others could make. For this function there were to less users that made crossword puzzles and to less was known about how one did this. I consider this as a disappointment.
There was much structure in the project. All stages were clearly described by the supervisor and I was insured of good support when I needed it. The software development method, user centered design is a very good method, especially for this program where the user requirements were not given. Much is written about user centered design and however not all those sources agree with each other the general line is the same. User centered design really involves the user and I was able to make the product better suited for users by using this method. It is also very helpful for me, as being the designer. User centered design gave me a good structure. Every phase is known, I knew what I was doing and why. The only thing problematic to this method is that users can deceive you. Users can say things that conflicts with each other or users can chose for 4 different things so the method makes it harder to take out a general line. When I spoke more to the users and ask for argumentation I could however see that the general line in this project is pretty the same. In this project users didn't say conflicting things and the general line could be distilled from their argumentation.

In this project I learned much about prototyping and how it is done. I learned much from resources and literature that I never saw before and was very useful. There is an overload of documentation but it is hard to find the correct ones. In this I was helped by the supervisor and he did a great job. User centered was a method I never used before. It is a very interesting method and it is great to see how it functions in a real situation. From this I also learned.

At last I hope that my work will be used in the future. That it may be valuable to others in every way. I hope that users will learn something from the experience I had, I did the best I could to describe those experiences and the development process as a whole.

References

Beyer H & Holtzblatt K (1998) Contextual Design: Defining Customer-Centered Systems. San Francisco, Morgan Kaufmann Publishers.

Biesterfeld, J. (2005) *Usability - Im Zweifel für den Anwender*. Retrieved February 2, 2006 from http://www.transline.de/transline-tecNews/usability-im-im-zweifel-fuer-den-anwender#1.

Cambon Jensen, S., 1997. *Design and implementation of crossword compilation programs using sequential approaches.* Master Thesis. Odense University.

Color Wheel Pro (2006). Retrieved March 9, 2006 from: http://www.color-wheel-pro.com/color-meaning.html

Constantine, L.L., and Lockwood, L. A. D. (1999). Software For Use: A Practical Guide to the Models and Methods of Usage-Centered Design. New York: ACM Press.

Cooper, A. (1999) *The inmates are running the asylum*. Indianapolis, SAMS McMillan.

Evans, M.K. (2002). Peter Merholz and Nathan Shedroff: User Centered Design. *Digital Web Magazine* [online]. Retreived February 7, 2006 from: http://www.digital-web.com/articles/peter_merholz_and_nathan_shedroff/

Greenbaum, J. and Kyng, M. (1991). *Design at Work: Cooperative Design of Computer Systems.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Hix, D. & Hartson, H.R. (1993). *Developing User Interfaces: Ensuring Usability Through Product & Process*. New York: John Wiley & Sons, Inc.

IBM (2006) retrieved April 19, 2006 from http://www-3.ibm.com/ibm/easy/eou_ext.nsf/Publish/2.

Ihlström, C. (2005). The e-newspaper innovation - converging print and online. Accepted to the International Workshop on Innovation and Media: Managing changes in Technology, Products and Processes, Stockholm.

iRex, (2006) retrieved 3 March 2006 from http://www.irextechnologies.com

ISO/IEC (1999) 13407 Human-centred Design Processes for Interactive Systems. ISO/IEC 13407: 1999 (E).

Kjeldskov, J. and Graham, C. (2003). A Review of Mobile HCI Research Methods" in *Proceedings of the 5th International Mobile HCI Conference*, L. Chittaro (Ed.), Udine, Italy, September 8-11, 2003, pp. 317-335.

Kristoffersen, S. and Ljunberg, F. (2000). Mobility: From stationary to mobile work in *Planet Internet*, Braa, K., Sørensen, B. and Dahlbom, B. (Eds.) Lund, Sweden: Studentlitteratur.

MacIntyre, R. Estep, K.W. & Sieburth, J.M. (1990). "Cost of User-friendly Programming". *Journal of Forth Application and Research*, 6, 2, 103-115.

Negroponte, N. (1995) *Being digital*. London: Hodder & Stoughton.

Nielsen, J. (1995) *Usability Engineering*. Mountain View, California, USA: AP Professional.

Norman, D.A. and Draper, S.W., (1986). *User Centered System Design*. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.

Preece, J., Rogers, Y., Sharp, H. (2002) *Interaction design: beyond human – computer interaction.* New York, NY, USA: John Wiley & Sons, Inc.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T. (1994) "an HCI design approach" in *Human Computer Interaction*. Harlow: Pearson Education Limited.

Rauch, T.L., Soderston, C., and Hill, G.W. (1996). Defining a User-Centered Design Process. *In ed.:STC Annual Conference Proceedings*. Research Ttriangle Park: International Business Machines Corporation, 341-344.

Silverman, D. (1999). Doing qualitative research: a practical handbook. Thousand Oaks, CA, USA: Sage Publications Ltd.

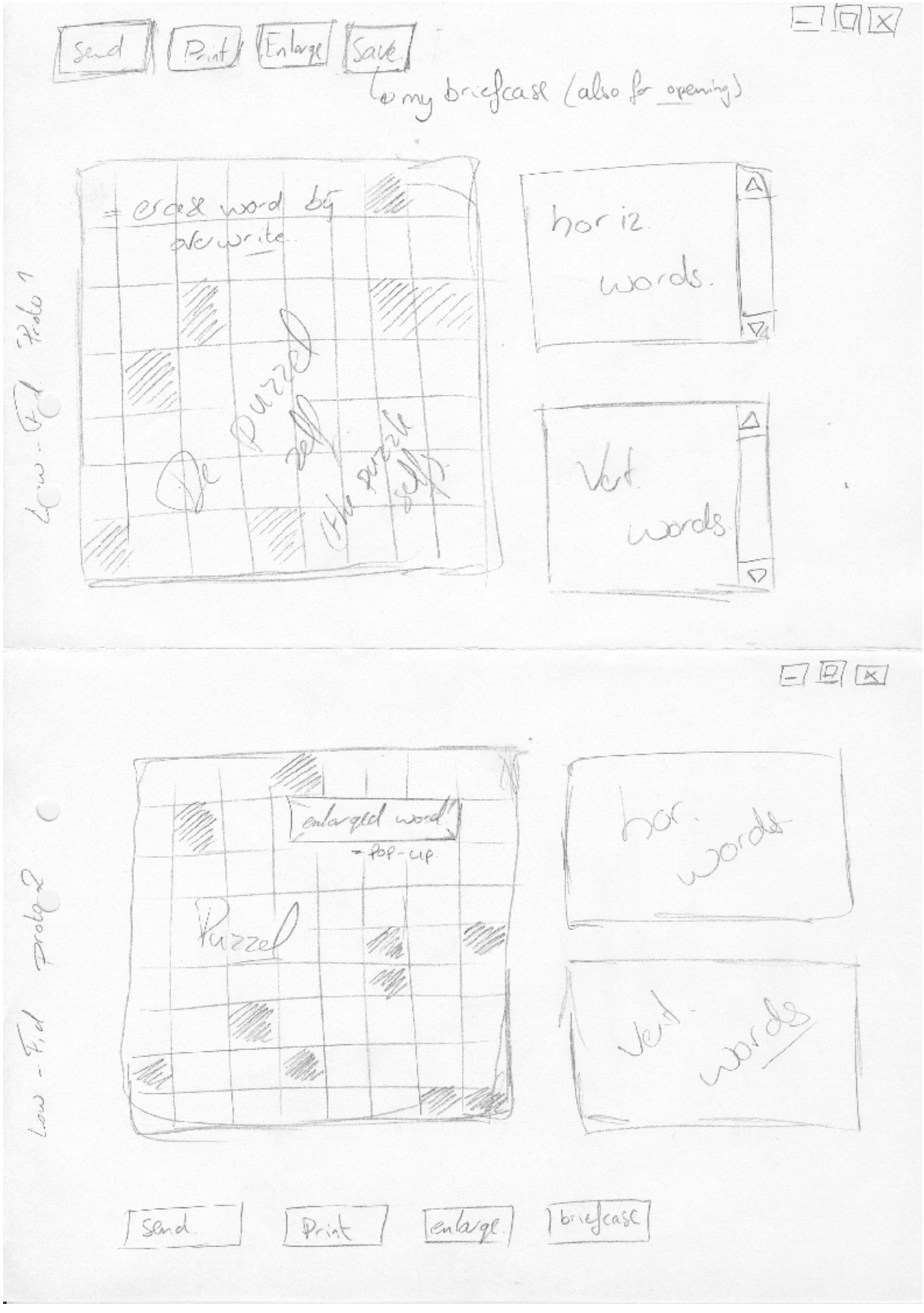
Sommerville, I. (2001) *Software Engineering.* (6th edition). Harlow: Pearson Education Limited.

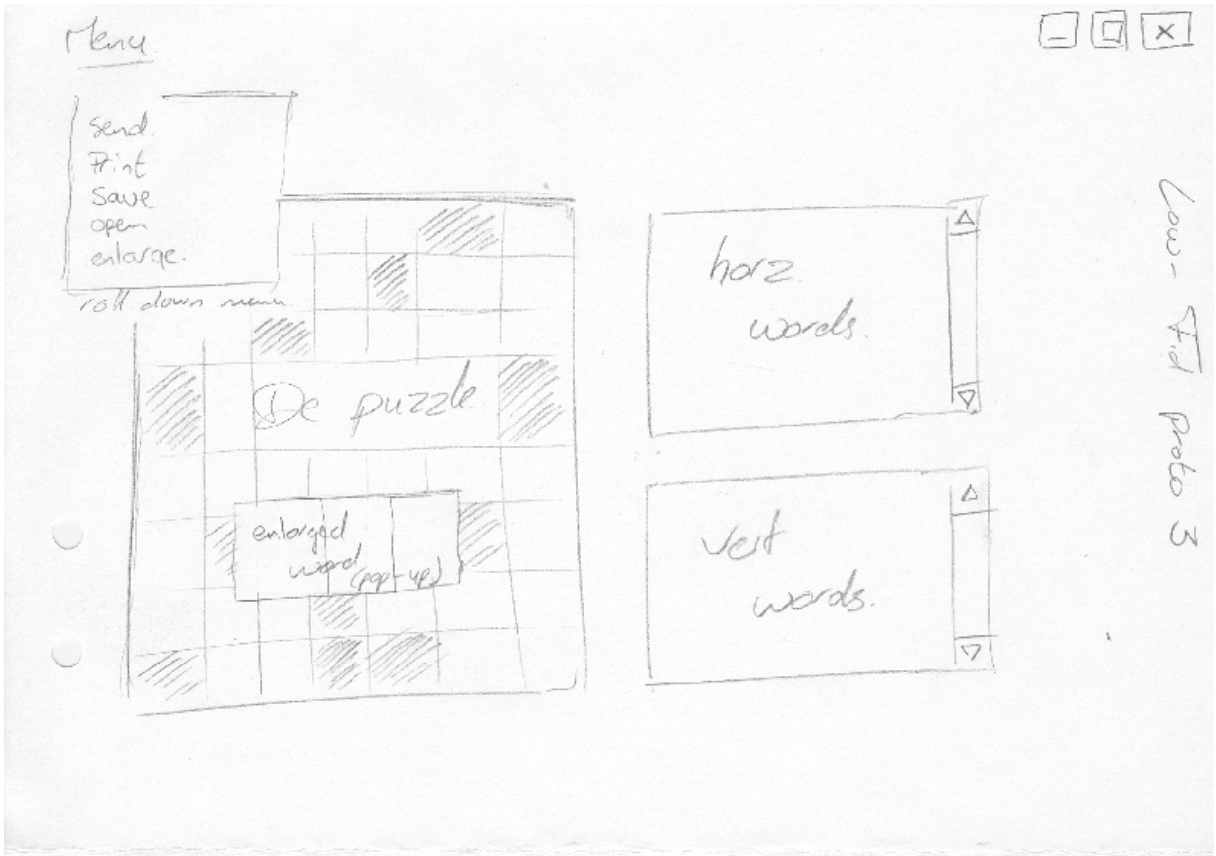
Sony, (2006) retrieved 3 March 2006 from http://www.sonystyle.com/is-bin/INTERSHOP.enfinity/eCS/Store/en/-/USD/SY_DisplayProductInformation-Start?CategoryName=&ProductSKU=PRS500&TabName=specs&var2=

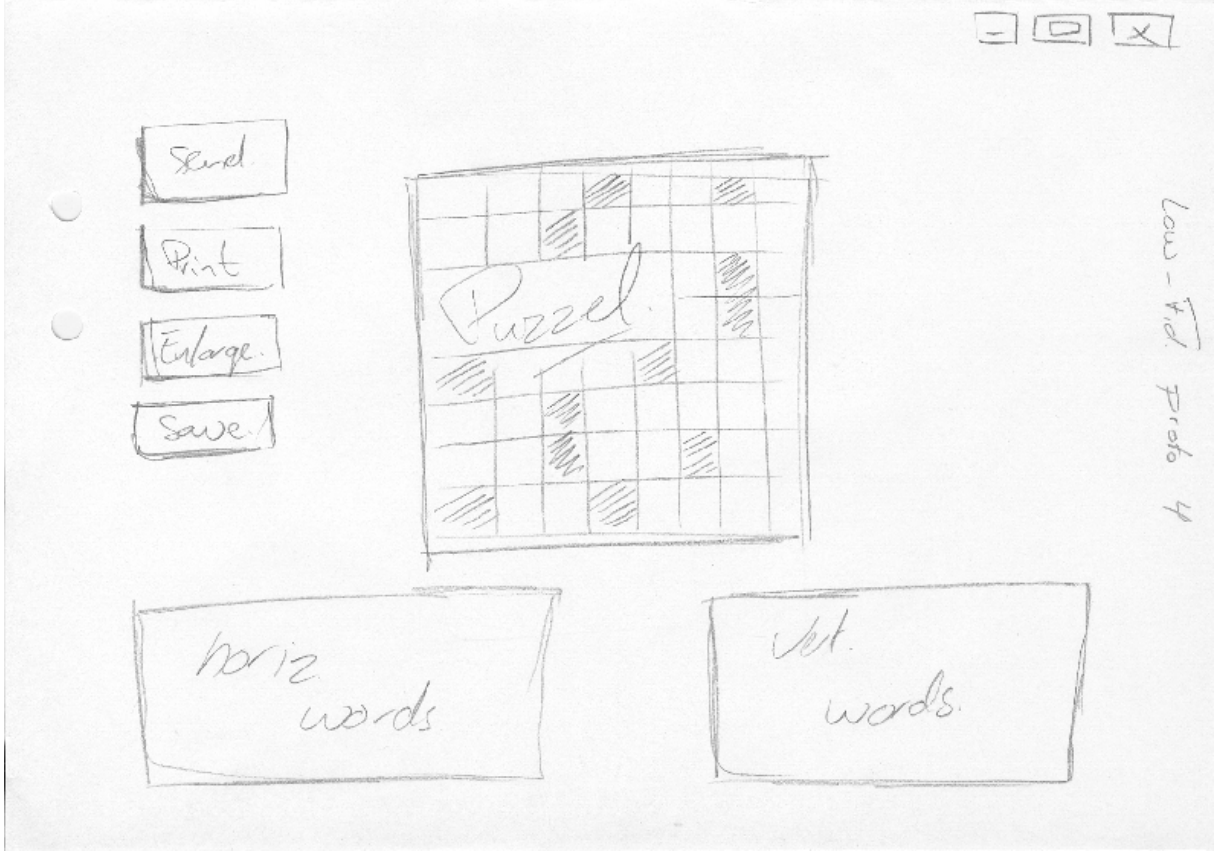
Ur Rehman, R. & Paul Ch. (2003) *The Linux Development Platform*. Upper Saddle River, New Jersey: Prentice Hall.

Wixon, D., Holtzblatt, K. and Knox, S. (1990). "Contextual Design: An Emergent View of System Design." CHI '90 Proceedings 329-336.

Appendix 1

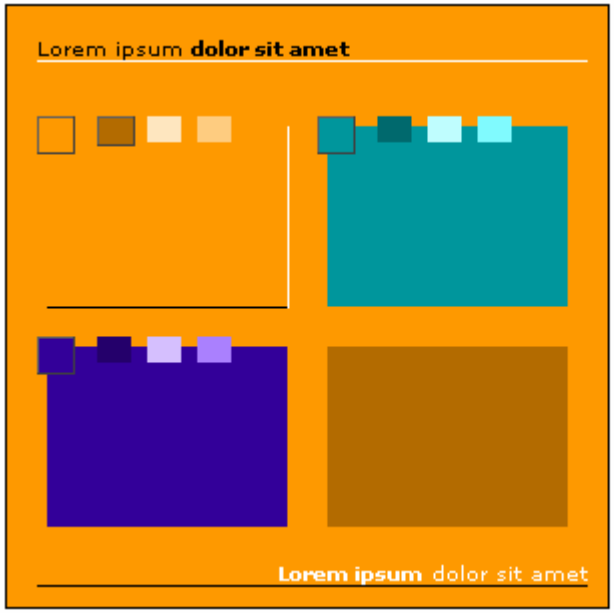

Send
Print
Enlarge
Save
to my briefcase (also for opening)
Low - Fid Proto 1
horiz. words.
Vert. words.
Low - Fid Proto 2
enlarged word
- pop-up
Puzzel
hor. words
vert. words
Send
Print
enlarge
briefcase


Menu
Send
Print
Save
open
enlarge.
roll down menu
De puzzle
enlarged word (opt-ye)
horz words.
vert words.
Low - Fid Probe 3

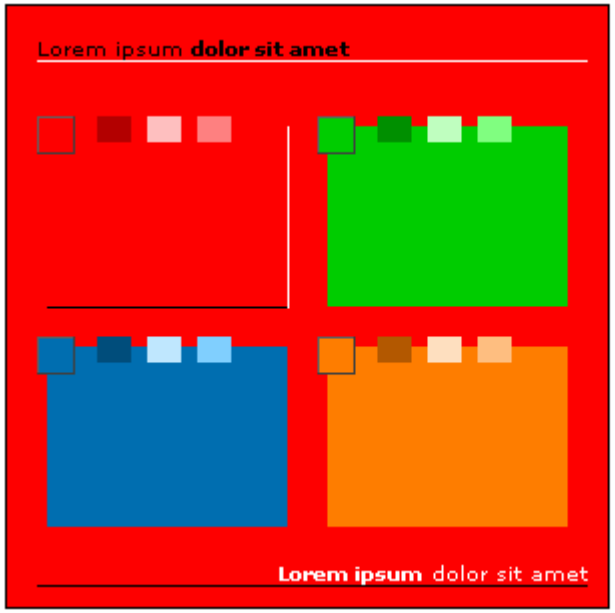

Send
Print
Enlarge
Save
Puzzel
horiz words
vert words.
Low - Fid Probe 4

#330099
#24006B
#D5BFFF
#AA80FF
#FFCC00
#B38F00
#FFF2BF
#FFE680
#D3FF00
#94B300
#F4FFBF
#E9FF80
#920099
#66006B
#FCBFFF
#F980FF

#0033CC
#00248F
#BFCFFF
#809FFF
#FF9900
#B36B00
#FFE6BF
#FFCC80
#FFE200
#B39E00
#FFF8BF
#FFF180
#3E0099
#2B006B
#D9BFFF
#B380FF

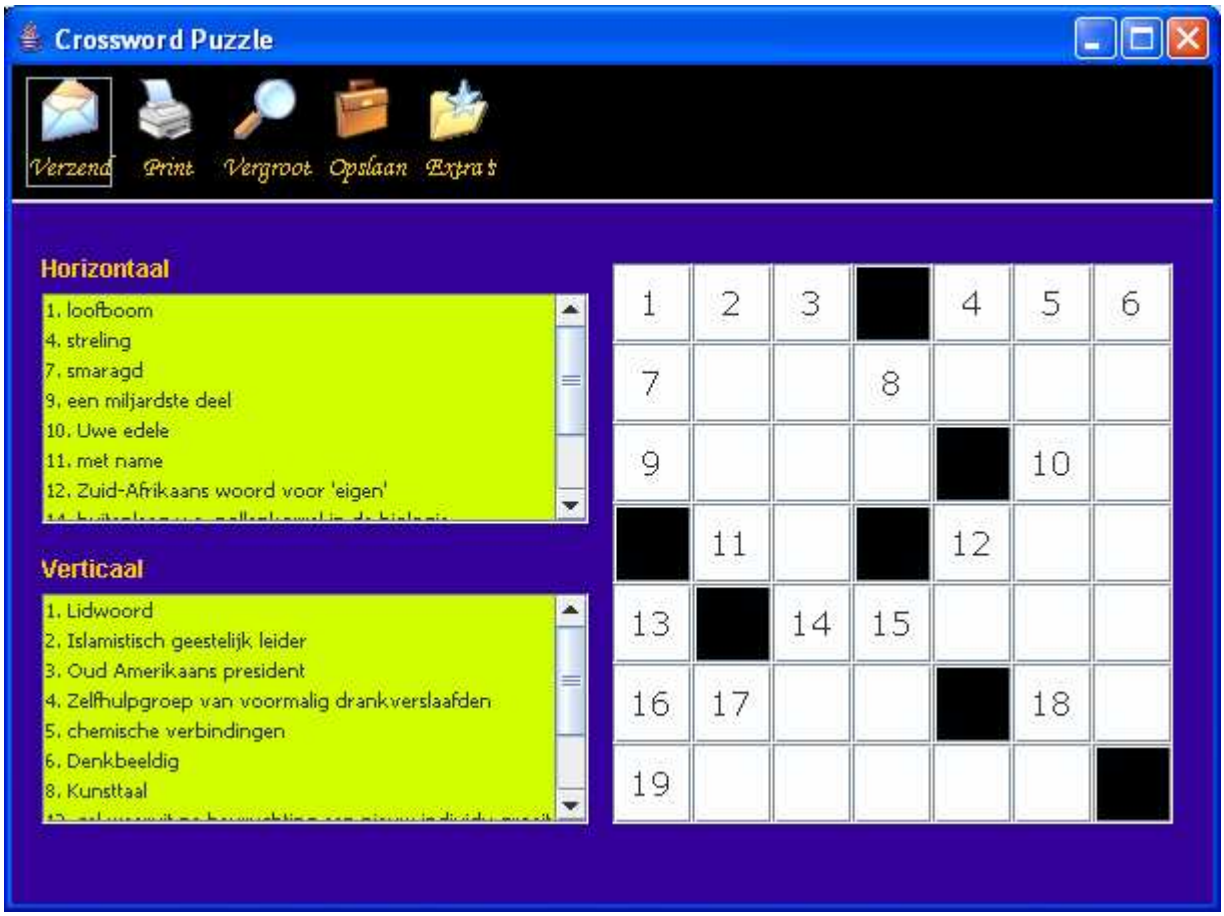


#FF9900
#B36B00
#FFE6BF
#FFCC80
#00969C
#00696D
#BFFDFF
#80FAFF
#330099
#24006B
#D5BFFF
#AA80FF

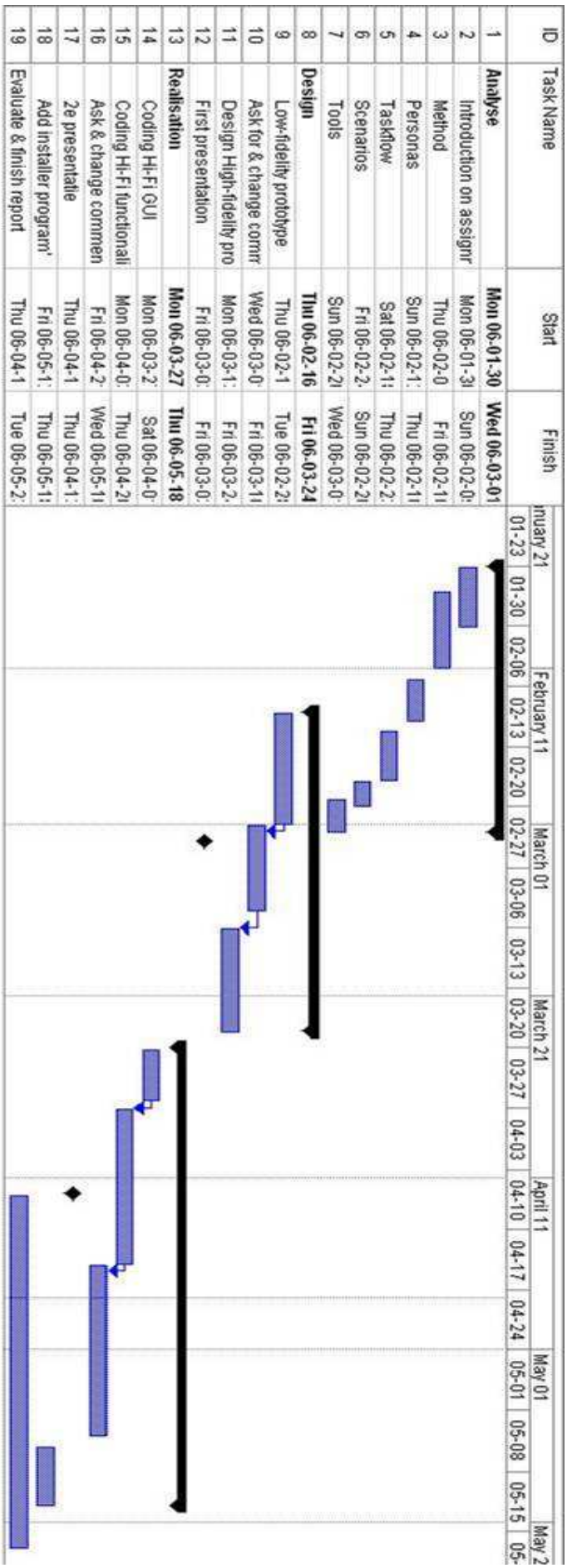


#FF0000
#B30000
#FFBFBF
#FF8080
#00CC00
#008F00
#BFFFBF
#80FF80
#006EB0
#004D7B
#BFE7FF
#80CFFF
#FF7D00
#B35800
#FFDFBF
#FFBE80

Appendix 2


Crossword Puzzle
Verzend
Print
Vergroot
Opslaan
Extra's
Horizontaal
1. loofboom
4. streling
7. smaragd
9. een miljardste deel
10. Uwe edele
11. met name
12. Zuid-Afrikaans woord voor 'eigen'
Verticaal
1. Lidwoord
2. Islamistisch geestelijk leider
3. Oud Amerikaans president
4. Zelfhulpgroep van voormalig drankverslaafden
5. chemische verbindingen
6. Denkbeeldig
8. Kunsttaal
1 2 3 4 5 6
7 8
9 10
11 12
13 14 15
16 17 18
19

Appendix 3



ID	Task Name	Start	Finish
1	**Analyse**	**Mon 06-01-30**	**Wed 06-03-01**
2	Introduction on assignr	Mon 06-01-3(	Sun 06-02-0(
3	Method	Thu 06-02-0	Fri 06-02-1(
4	Personas	Sun 06-02-1(	Thu 06-02-1(
5	Taskflow	Sat 06-02-1(	Thu 06-02-2(
6	Scenarios	Fri 06-02-2(	Sun 06-02-2(
7	Tools	Sun 06-02-2(	Wed 06-03-0(
8	**Design**	**Thu 06-02-16**	**Fri 06-03-24**
9	Low-fidelity prototype	Thu 06-02-1	Tue 06-02-2(
10	Ask for & change comr	Wed 06-03-0	Fri 06-03-1(
11	Design High-fidelity pro	Mon 06-03-1(	Fri 06-03-2(
12	First presentation	Fri 06-03-0(	Fri 06-03-0(
13	**Realisation**	**Mon 06-03-27**	**Thu 06-05-18**
14	Coding Hi-Fi GUI	Mon 06-03-2(	Sat 06-04-0(
15	Coding Hi-Fi functionali	Mon 06-04-0(	Thu 06-04-2(
16	Ask & change commen	Fri 06-04-2(	Wed 06-05-1(
17	2e presentatie	Thu 06-04-1	Thu 06-04-1(
18	Add installer program'	Fri 06-05-1(	Thu 06-05-1(
19	Evaluate & finish report	Thu 06-04-1	Tue 06-05-2(